

Classificação de Objetos usando os momentos de Zernike

José Ícaro Santana Bernardes, Pedro Renoir Silveira Sampaio, Kamila Amélia Sousa Gomes,
Iális Cavalcante de Paula Júnior, Marcelo Marques Simões de Souza

¹Campus de Sobral – Universidade Federal do Ceará (UFC)
Caixa Postal 62010-560 – Sobral – CE – Brazil

stnicaro@gmail.com, renoir@alu.ufc.br, fariaas.kamila@gmail.com,
ialis@sobral.ufc.br, marcelo.mssouza@gmail.com

Abstract. *The Zernike moments are sets of complex polynomials used in computer vision problems and there are several works that explore as a descriptor properties of attributes that a rotation is invariable. In this article, the performance of Zernike moments is explored in a classification problem to which the sets of each class have a variety of possibilities. For this, the Natural Images database and three machine learning algorithms were used. With the intention of bringing a comparison and showing the impact of the variation of parameters on the performance of the algorithms. At the end of the work, the best result was published at 77 % accuracy.*

Resumo. *Os momentos de Zernike são conjuntos de polinômios complexos usados em problemas de visão computacional e existem diversos trabalhos que exploram as suas propriedades descritor de atributos que se apresenta invariável a rotação. Neste artigo, é explorado o desempenho dos momentos de Zernike em um problema de classificação ao qual os conjuntos de cada classe possuem uma variedade de perspectivas. Para isso, foi utilizado a base de dados Natural Images e três algoritmos de aprendizado de máquina. Com o intuito de trazer uma comparação e mostrar os impactos das variações dos parâmetros nos desempenhos dos algoritmos. Ao final do trabalho, o melhor resultado obtido foi o de 77% de acurácia.*

1. Introdução

A classificação de imagens de objetos é um clássico problema da área de visão computacional. Atualmente, na literatura existem trabalhos que apresentam diferentes abordagens sobre tal temática. Dentre as técnicas de classificação existentes as que mais se destacam são as que se baseiam em cor, textura e principalmente na forma da imagem.

Os descritores baseados em forma são divididos em duas subcategorias: a de similaridade baseada em contorno e a de similaridade baseada em região [Cruz 2004]. A diferença entre elas é que os descritores baseados em contorno esboçam propriedades relacionados ao contorno da forma, enquanto as que se baseiam em região usam distribuição dos pixels ao longo de diferentes regiões da imagem [Cruz 2004].

Este artigo foca nos descritores de forma baseado em região, mais especificamente usando os momentos de Zernike. Esses momentos possuem a propriedade de serem invariantes a rotação [Khotanzad and Hong 1990]. Nesse contexto, o trabalho de

[Cecotti 2020] traz um estudo comparando a robustez de diversos descritores, dentre eles destaca-se os momentos de Zernike, que possuem essa propriedade em problema de identificação da morfologia de galáxias. Além disso, também é feito um estudo para classificação de imagens de galáxias em dois tipos: espirais e elípticas.

Outra abordagem comum na literatura é a identificação de objetos. Nesse sentido, o trabalho de [Carvalho and Fernandes 2020] faz um estudo comparativo entre as técnicas de momentos de Hu, momentos de Zernike e uma arquitetura de VGG (*Visual Geometry Group*) para identificar e verificar a presença de um determinado objeto em uma imagem, no caso uma chave.

Esse artigo explora a capacidade dos momentos de Zernike em identificar objetos que pertencem à mesma classe, porém possuem uma distribuição espacial dos pixels diferentes, diferença essa que muitas vezes são geradas por conta das capturas de imagens terem sido feitas por diferentes perspectivas ou estavam em diferentes condições durante as capturas dessas imagens. Tendo em vista tal problemática, as análises feitas neste trabalho tem como objetivo trazer resultados que agreguem conhecimento sobre a robustez do momentos de Zernike como extrator de características.

Na seção 2 será mostrado os materiais e os algoritmos utilizados no trabalho. Na seção 3 será descrita a metodologia aplicada para obter os resultados, que serão mostrados na seção 4. Por fim, na seção 5 a conclusão e a discussão acerca do objetivo do trabalho.

2. Materiais e Métodos

2.1. Momentos de Zernike

Os polinômios de Zernike são um conjunto de polinômios complexos que forma um conjunto ortogonal complexo no interior de um círculo unitário [Cruz 2004]. Para calcular o momentos de Zernike em uma Figura, o centro da imagem é considerado como origem e as coordenadas dos pixels são mapeadas para o intervalo do círculo, isto é, $x^2 + y^2 \leq 1$ [Cruz 2004]. Em que x e y são as coordenadas do pixel e 1 representa o raio desse círculo. Um outro valor para o raio também poderia ser definido. Esse círculo delimita a região onde será feita o cálculo dos momentos de Zernike. Este gera dados a respeito da forma como os pixels estão distribuídos na imagem. Os pixels que ficam fora deste círculo são descartados do cálculo [Cruz 2004]. Isso tem impacto direto na extração de características já que informações importantes podem ficar de fora da análise.

Os momentos de Zernike possuem a característica de ser invariante a rotação [Khotanzad and Hong 1990]. A invariância a escala e translação podem ser obtida deslocando e escalando a imagem antes de realizar o cálculo [Cruz 2004]. Para obter a invariância à translação, basta mover a origem para o centróide antes do cálculo dos momentos [Cruz 2004]. Já em relação a escala pode ser obtida aumentando ou reduzindo a forma fazendo com que m_{00} seja igual ao tamanho pré-determinado da imagem. Em que m_{00} corresponde a um momento geométrico da imagem [Cruz 2004].

2.2. Base de Dados

A base de dados escolhida foi a “Natural Images” disponível na plataforma Kaggle [Kaggle 2018]. Essa base possui 6899 imagens e 8 classes distintas. Essas classes são “avião” (727 imagens), “carro” (968 imagens), “gato” (885 imagens), “cachorro” (702

imagens), “flores” (843 imagens), “frutas” (1000 imagens), “motocicleta” (788 imagens) e “pessoas” (986 imagens).

2.3. k-vizinhos mais próximos

O algoritmo de k-vizinhos mais próximos (k-NN) é um classificador de aplicação simples. Ele busca os registros que possuem valores semelhantes, usando alguma métrica de distância, que funciona para definir a classe majoritária a qual é atribuída a esse registro [Bruce and Bruce 2019].

A semelhança de amostras é determinada com base em alguma métrica de distância. Existem diversas técnicas para medir distância, tais como: distância Euclidiana e a distância de Manhattan. A distância Euclidiana é definida na Equação 1. Em que (x_1, x_2, \dots, x_n) e (y_1, y_2, \dots, y_n) representam as coordenadas dos vetores \vec{X} e \vec{Y} , respectivamente.

$$\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} \quad (1)$$

Já em relação a distância de Manhattan é definida como a distância entre dois pontos traçada em uma única direção de cada vez [Bruce and Bruce 2019], como é descrita na Equação 2.

$$|x_1 - y_1| + |x_2 - y_2| + \dots + |x_n - y_n| \quad (2)$$

Outro fator que impacta nos resultados do k-NN é o dimensionamento dos dados. Dados com escala de valores muito grande em relação a outros tipos de dados dominarão a medida. Para solucionar esse problema é preciso que os dados passem por um processo de padronização, também chamado de normalização [Bruce and Bruce 2019], antes da aplicação no algoritmo. A técnica mais comum é a z-score [Bruce and Bruce 2019] que consiste na aplicação da Equação 3. Em que w é uma coordenada, \bar{w} é a média dos valores nesse eixo e s o desvio padrão e z o valor normalizado.

$$z = \frac{w - \bar{w}}{s} \quad (3)$$

Com isso, impedimos que o impacto de uma variável seja maior do que de outra por sua escala.

2.4. Perceptron Multicamadas

O Perceptron multicamadas (MLP) é uma rede neural derivada da Perceptron simples. Ele é composto por uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída.

O treinamento do MLP funciona da seguinte forma, o algoritmo de retropropagação primeiro faz uma previsão, depois o erro é calculado através da diferença entre o valor esperado e o valor obtido da rede. Após essa etapa, o erro passa por cada camada no sentido contrário ao que veio para medir a contribuição do erro em cada conexão, em seguida é feito o ajuste dos pesos para reduzir o erro através de um método de otimização [Geron 2019], sendo o Gradiente Descendente o mais famoso.

Dentre os fatores que influenciam no desempenho da MLP se destacam a quantidade de camadas ocultas e de neurônios por camada oculta. A quantidade de camadas ocultas varia de problema para problema, mas uma técnica que ajuda a encontrar esse número é expandir a rede até que comece a sobreajustar o conjunto de treinamento [Geron 2019]. Para o número de neurônios, também varia dependendo do caso, pode ser feito o mesmo processo para a camada oculta ou deixar todas as camadas com a mesma quantidade, ou ainda distribuir quantidades diferentes de neurônios, dimensionando como um funil com cada vez menos neurônios por camada [Geron 2019].

Além disso, ainda tem a taxa de aprendizagem (η). Este exprime o quão rápido o processo de treinamento está convergindo para o menor erro [Ivan Nunes da Silva and Flauzino 2016]. O valor de η deve ser escolhido com cuidado para evitar instabilidades no processo de treinamento, sendo adotados valores que ficam no intervalo $0 < \eta < 1$.

2.5. Máquina de Vetores de Suporte

Máquina de Vetores de Suporte (SVM) é um modelo muito poderoso e versátil de Aprendizado de Máquina capaz de realizar classificação lineares ou não lineares, de regressão e detecção de valores atípicos [Geron 2019]. O objetivo do classificador SVM está baseado em uma busca de um hiperplano ótimo [Bonesso 2013], ou seja aquela linha ou fronteira de maior separação dos dados.

Um fator que impacta no desempenho do SVM é o tipo de classificador que usamos, que podem ser: com margens rígidas ou com margens suaves. A margem é a distância entre os dois hiperplanos, podendo ter margens mais largas ou mais estreitas. Essa largura é determinada pelo parâmetro C . Se o valor de C aumenta a largura diminui, caso valor de C diminua a largura aumenta. O SVM com margens rígidas só funciona em problemas linearmente separáveis e é bastante sensível a valores atípicos [Geron 2019]. Já o SVM com margens suaves é indicado para problemas não lineares, já que mais flexível permitindo algumas violações de margem e até amostras do lado errado do hiperplano, seguindo um parâmetro de penalização [Bonesso 2013].

Outra questão importante no SVM é o *Kernel*. O Truque do *Kernel* [Bonesso 2013] possibilita que o espaço original seja mapeado em um espaço de produto escalar de alta dimensão chamado espaço de característica [Bonesso 2013]. Através desse mapeamento dos dados de entrada, os quais no espaço original não possuem fronteiras linearmente separáveis, em um espaço de dimensões mais altas aumentam as chances de ser linearmente separáveis [Bonesso 2013].

3. Metodologia

A primeira parte do trabalho consiste no tratamento da imagem. Nessa parte a imagem passa por um processo de segmentação por binarização, que significa comparar o valor dos pixels ao limiar. Os pixels que apresentam valor maior que esse limiar são definidos como objetos de interesse. Para contornar possíveis problemas de imagens com algum grau de variação de luminosidade, que pode estar presente em algumas imagens e acabar atrapalhando no tratamento da mesma em relação a definição dos seus pixels, optou-se por usar a binarização adaptativa que calcula diferentes valores de limiar para diferentes regiões da imagem. Posteriormente, foi aplicada uma operação morfológica de erosão a

fim de diminuir os ruídos gerados pela binarização, tornando assim os resultados mais uniformes. Podemos ver na Figura 1 um exemplo de imagem de entrada (a) e a sua transformação após o processo de pré-processamento (b).

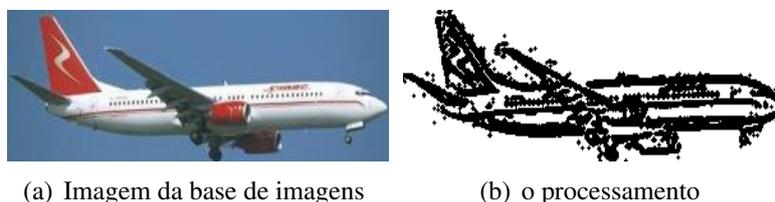


Figura 1. Exemplo de aplicação do pré-processamento . Fonte: Elaborada pelo autor.

Após efetuar este tratamento, as imagens seguem para o cálculo dos momentos de Zernike. Esse cálculo é feito definindo o valor do raio dos círculos que é um parâmetro a ser determinado pelo autor dos experimentos. Nesse caso, ficou decidido que o máximo valor entre as dimensões das imagens seria a medida do raio, com o intuito de analisar a distribuição dos pixels na maior região possível. E assim, os resultados são agrupados formando uma matriz de características que será usada no processo de classificação.

No processo de classificação as classes foram representadas como números e ficaram da seguinte maneira: a classe “avião” como número 1, “carro” como número 2, “gato” como número 3, “cachorro” como número 4, “flores” como número 5, “frutas” como número 6, “motocicleta” como número 7 e “pessoas” como número 8.

A próxima etapa foi a aplicação da matriz de características em algoritmos de classificação, são eles: k- vizinhos mais próximos (k-NN), Máquina de vetores de suporte (SVM) e Perceptron multicamadas (MLP). Foi então utilizado o k-fold, como uma estratégia de validação cruzada para obter os desempenhos, com o valor de k variando de 2 a 8. Como não teve nenhuma diferença de rendimento discrepante para diferentes valores de k, os resultados mostrados são para k igual a 8. O desempenho foi calculado como a média da acurácia que foi obtida ao final do processo de treino e teste em cada parte. Já a acurácia é a quantidade de amostras classificadas corretamente pela quantidade total de amostras. Ela foi escolhida por ter um fácil entendimento e traz uma informação muito relevante pra quem procura uma técnica de classificação de objetos. A Figura 2 resume as etapas do processo aplicado no trabalho.



Figura 2. Fluxograma da metodologia aplicada. Fonte: Elaborada pelo autor.

4. Resultados

Os experimentos feitos foram divididos em 2 partes: momentos de Zernike e momentos de Zernike com a aplicação de z-score aos dados calculados.

4.1. k-NN

Nos testes foram usados diferentes valores de k vizinhos, variando de 2 a 7. Como não apresentaram nenhuma diferença significativa, os resultados mostrados são apenas para k igual a 2 e 7. As métricas de distância usadas foram Euclidiana e Manhattan. Os dados também foram comparados com e sem a aplicação de z-score.

As métricas Euclidiana e de Manhattan não apresentam diferenças muito significativas, obtendo os mesmo resultado, considerando os melhores obtidos para cada métrica. Os testes em que os dados foram submetidos ao z-score também mostraram melhora nos resultados do que os dados que não foram. Os resultados obtidos seguem na Tabela 1.

Tabela 1. Resultados experimentais com classificação baseada em k-NN.

Métrica	k-vizinhos	z-score	Acurácia no treino	Acurácia no teste
Euclidiana	2	sem	85%	68%
Euclidiana	2	com	86%	70%
Euclidiana	7	sem	78%	71%
Euclidiana	7	com	80%	73%
Manhattan	2	sem	85%	69%
Manhattan	2	com	86%	70%
Manhattan	7	sem	78%	72%
Manhattan	7	com	79%	73%

4.2. MLP

No MLP a função de ativação foi a tangente hiperbólica. Os experimento foram conduzido variando a quantidade de camadas ocultas, a quantidade de neurônios nas camadas ocultas e a taxa de aprendizagem (η).

Os dados com a aplicação z-score provocaram uma melhora na acurácia em relação aos dados sem z-score. Valores muito altos ou menores da taxa de aprendizagem reduzem a acurácia. Com o aumento de camadas e dos neurônios fez com o MLP acabe sendo ajustado aos dados de treinamento. A Tabela 2 mostra os resultados obtidos.

Tabela 2. Resultados experimentais com classificação baseada em MLP.

N ^a de Camadas Ocultas	N ^a de neurônios	η	z-score	Acurácia no treino	Acurácia no teste
1	50	0.001	sem	78%	76%
1	50	0.05	sem	75%	74%
1	50	0.005	sem	76%	75%
2	(50,50)	0.001	sem	77%	75%
2	(50,50)	0.001	com	85%	77%
4	(50,50,50,50)	0.001	sem	77%	75%
4	(30,30,30,30)	0.001	com	85%	77%
4	(80,80,80,80)	0.001	com	100%	76%

4.3. SVM

O *Kernel* escolhido para o SVM foi o polinomial. Os parâmetros avaliados foram o grau do polinômio e o hiperparâmetro C .

O z-score nesse caso teve efeito contrário no SVM em relação aos outros classificadores, ou seja a acurácia diminuiu. Altos valores para C melhoram a acurácia. A Tabela 3 mostra os principais resultados.

Tabela 3. Resultados experimentais com classificação baseada em SVM.

Grau do Polinômio	C	z-score	Acurácia no treino	Acurácia no teste
12	1	sem	80%	74%
12	1	com	55%	49%
20	1	sem	84%	73%
8	1	sem	76%	74%
8	0.25	sem	74%	72%
8	50	sem	82%	76%
4	50	sem	78%	76%

4.4. Discussão dos resultados

A Tabela 4 compara o melhor resultado obtido neste trabalho com os melhores obtidos em [Carvalho and Fernandes 2020] para momentos de Hu e para os momentos de Zernike. Com isso, percebemos que os momentos de Zernike conseguem formar padrões distintos em diferentes contextos de trabalhos melhores que os momentos de Hu, que também são descritores baseado em região. Também é visto que os momentos de Zernike tendem a ter melhores resultados em classificações com várias classes do que em classificações binárias.

Tabela 4. Comparação de Técnicas.

Técnicas	Resultados
Momentos de Zernike	77%
Momentos de Hu	61%
Momentos de Zernike de [Carvalho and Fernandes 2020]	64%

Outro ponto que vale destacar é o fato do SVM, segundo [Geron 2019], ser um algoritmo sensível a escala das características, apresentando comportamento melhor quando os dados têm escalas padronizadas. Todavia, houve queda no desempenho do algoritmo quando os momentos de Zernike passam pela normalização z-score, como mostra a Tabela 3. Nessa Tabela, vemos por exemplo, a acurácia cair de 74% para 49% em um dos testes.

5. Conclusão

O trabalho consistiu na aplicação dos momentos de Zernike para classificação de objetos, passando por uma etapa de pré-processamento das imagens, construção da matriz de característica até chegarmos a etapa de coleta de resultados. Conforme mostrado pelas Tabelas no tópico de resultados, o SVM e MLP obtiveram melhores valores de resposta, já o k-NN ficou um pouco abaixo dos anteriores. O MLP foi o algoritmo que mostrou melhor resultado em acurácia.

O objetivo de trazer novos conhecimentos sobre os momentos de Zernike foi alcançado. Os resultados obtidos mostram que mesmo havendo uma grande variedade

na distribuição espacial dos pixels dentro de imagens pertencentes à mesma classe, os momentos de Zernike conseguiram discernir muito bem os objetos. Isso mostra que são ferramentas promissoras para classificar objetos independentemente das variações durante a obtenção das imagens. Essas informações são importantes visto que ainda não tinham muitos trabalhos explorando somente os momentos de Zernike como extrator de características.

A etapa de ajuste dos classificadores foi a mais complicada. Fatores como a limitação da máquina, onde foi feita a coleta dos resultados, e a quantidade de parâmetros a serem testados atrapalharam a obtenção de resultados mais próximos do ideal. A vantagem da técnica se dá por sua fácil reprodução. Isto ocorre graças ao fato de já existirem bibliotecas que trazem ferramentas prontas e já validadas para efetuar os cálculos.

Como ideia para a continuação em tal pesquisa, fica a possibilidade de trabalhar com a correlação dos momentos de Zernike, fazendo uma melhor combinação de atributos que aumentem a acurácia, bem como a variação em alguns tipos de atributos para aplicação de classificadores.

Referências

- Bonesso, D. (2013). Estimação dos parâmetros do kernel em um classificador svm na classificação de imagens hiperespectrais em uma abordagem multiclasse. Master's thesis, Universidade Federal do Rio Grande do Sul. Centro Estadual de Pesquisas em Sensoriamento Remoto e Meteorologia.
- Bruce, P. and Bruce, A. (2019). *Estatística Prática para Cientistas de Dados*. Alta Books, 1th edition.
- Carvalho, B. and Fernandes, B. (2020). Método de reconhecimento e padronização de captação para imagens de chave em protocolos de manutenção de estação radio base. *Revista de Engenharia e Pesquisa Aplicada*, pages 84–90.
- Cecotti, H. (2020). Rotation invariant descriptors for galaxy morphological classification. *International Journal of Machine Learning and Cybernetics*.
- Cruz, L. R. (2004). Reconhecimento de padrões em imagens por descritores de forma. Master's thesis, Universidade Federal do Rio Grande do Sul. Instituto de Informática. Programa de Pós-Graduação em Computação.
- Geron, A. (2019). *Mãos à obra: Aprendizagem de Máquina Scikit-Learn e TensorFlow*. Alta Books, 1th edition.
- Ivan Nunes da Silva, D. H. S. and Flauzino, R. A. (2016). *Redes Neurais Artificiais Para Engenharia e Ciências Aplicadas: Fundamentos Teóricos e Aspectos Práticos*. Artliber, 2th edition.
- Kaggle (2018). Natural images. Disponível em: <https://www.kaggle.com/prasunroy/natural-images>. Acessado em : 26 jun 2020.
- Khotanzad, A. and Hong, Y. (1990). Invariant image recognition by zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497.